

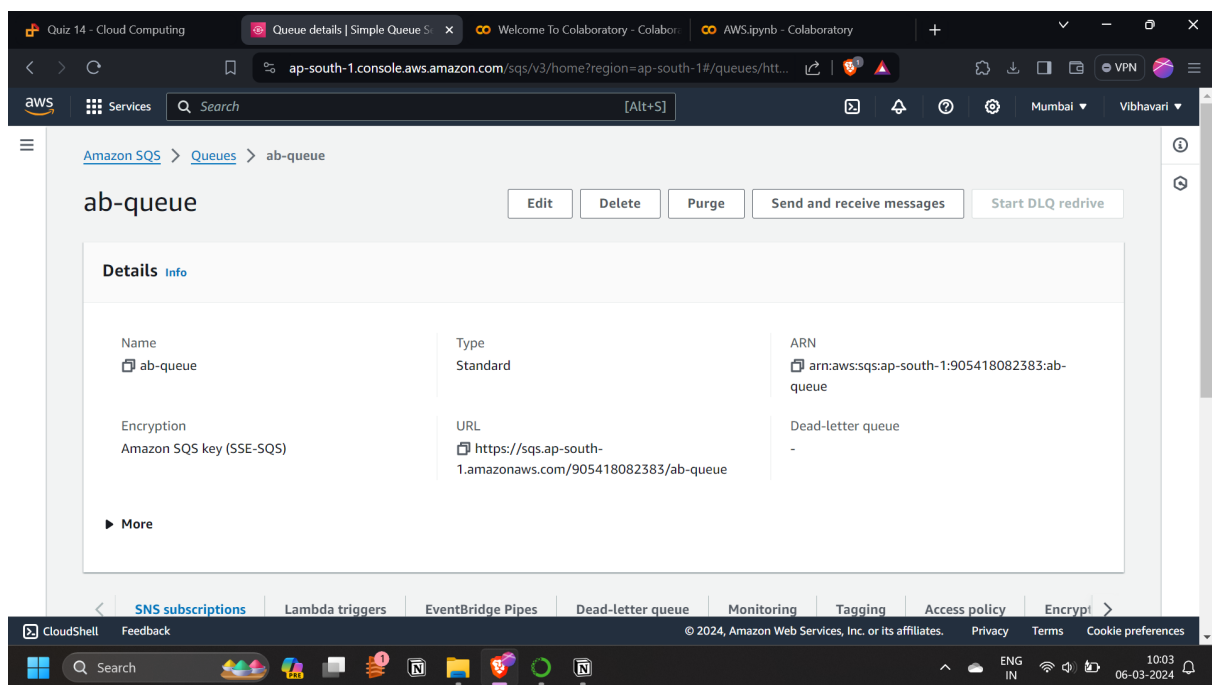
Project

Problem statement: Create a vendors order management system using AWS

AWS Services used:

- SQS
- Lambda function
- Dyanmodb
- IAM Role

Step1 : Create SQs Queue



SQS Template

```
AWSTemplateFormatVersion: '2010-09-09'
```

```
Description: Create SQS Queue
```

```
Resources:
```

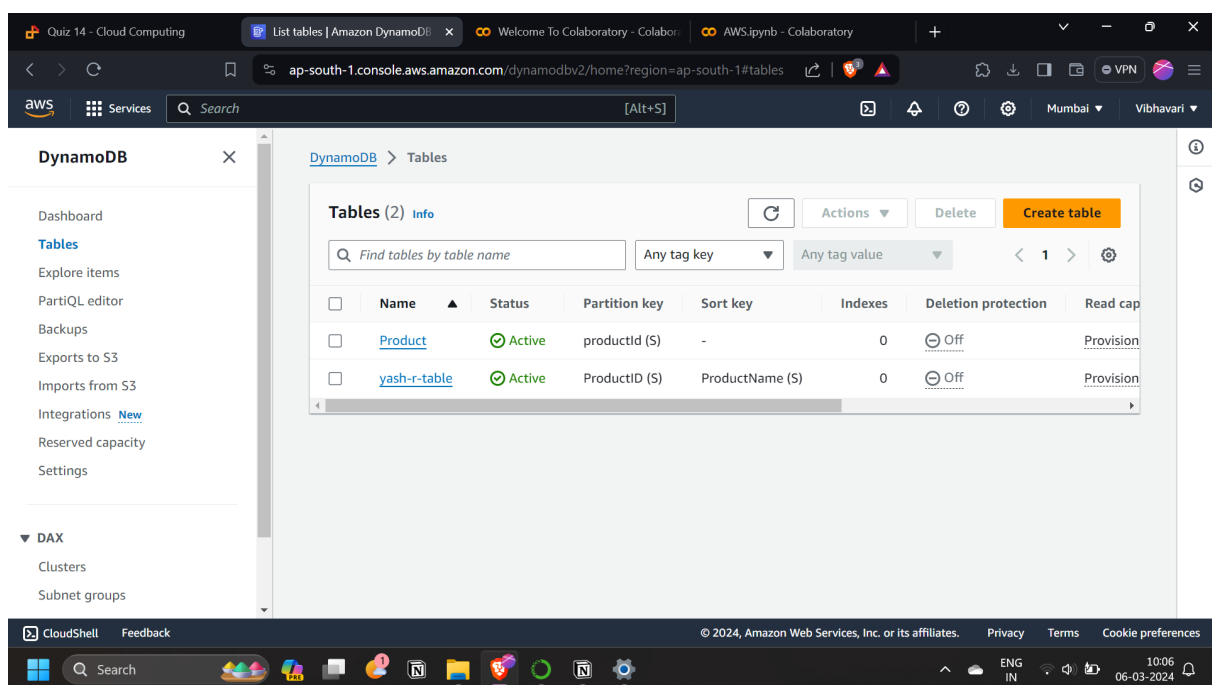
```
  MyQueue:
```

```
    Type: AWS::SQS::Queue
```

```
    Properties:
```

```
      QueueName: ab-queue
```

Step 2: Create Dynamodb table



Dynamodb table template

```
Resources:
```

```
  ProductTable:
```

```
    Type: 'AWS::DynamoDB::Table'
```

```
    Properties:
```

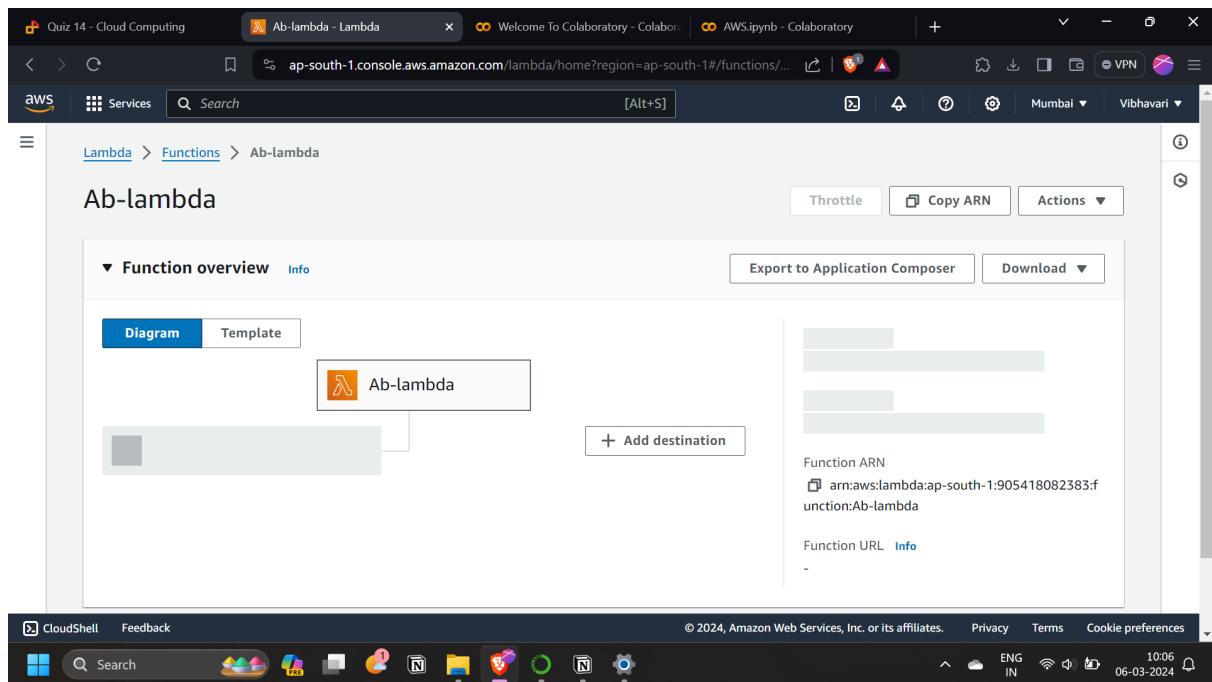
```
      TableName: Product
```

```
      AttributeDefinitions:
```

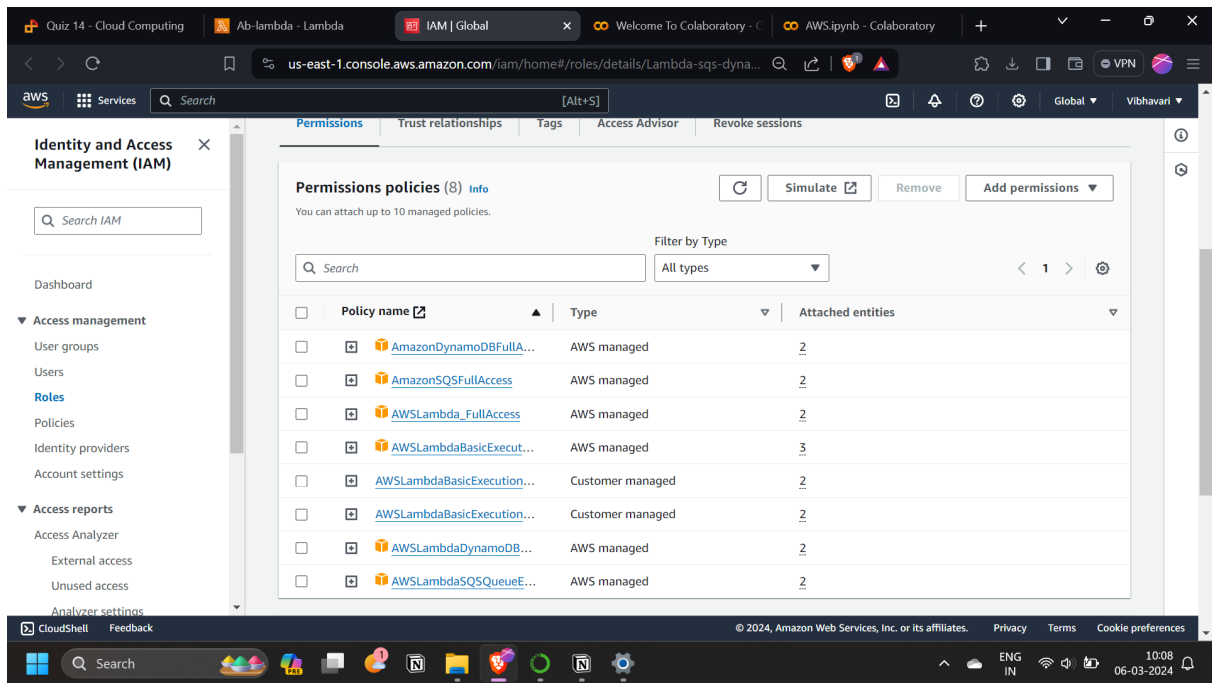
```
        - AttributeName: productId
```

```
AttributeType: S
KeySchema:
  - AttributeName: productId
    KeyType: HASH
ProvisionedThroughput:
  ReadCapacityUnits: 5
  WriteCapacityUnits: 5
```

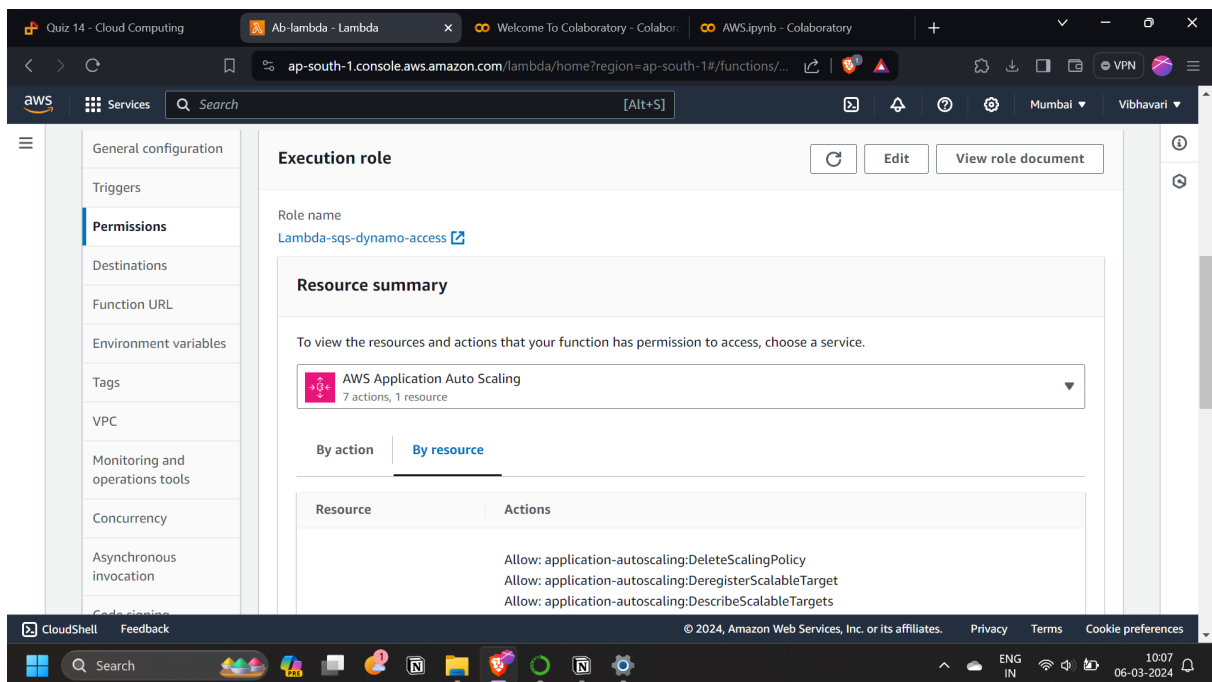
Step 3: Create Lambda function



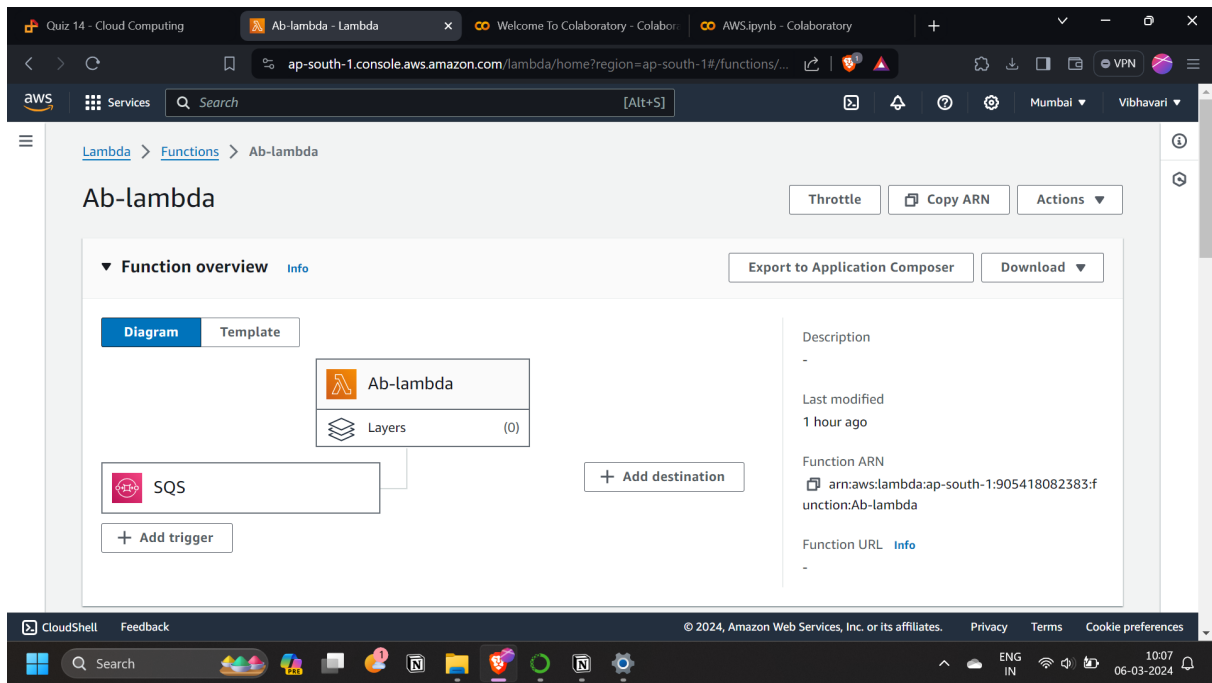
Step 4: Create IAM role



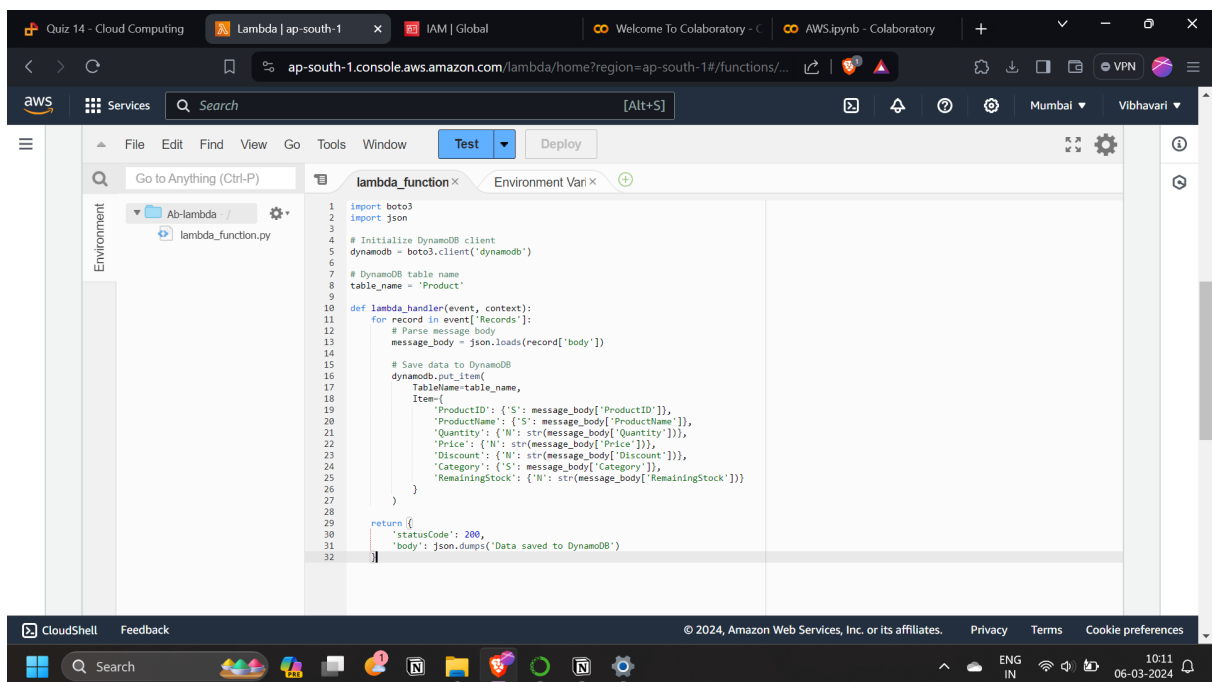
Step 5: Attached Role to Lambda function



Step 7: Added SQS Trigger to Lambda function



Step 7: Deployed code to Lambda to send code to Dynamodb



Step 8: Run python script to send the data to SQS

```

import boto3

# Set up SQS client
sqs = boto3.client('sqs', region_name='ap-south-1',aws_access.

# Define your SQS queue URL
queue_url = 'https://sqs.ap-south-1.amazonaws.com/90541808238

# Define the message data you want to send
message_data = {
    'ProductID': '12345',
    'ProductName': 'Sample Product',
    'Quantity': 10,
    'Price': 50,
    'Discount': 5,
    'Category': 'Sample Category',
    'RemainingStock': 100
}

# Send message to SQS queue
response = sqs.send_message(
    QueueUrl=queue_url,
    MessageBody=str(message_data)
)

# Print response
print(response)

```

Browser tabs: xcelerator - Learn By Doing, CloudFormation | ap-south-1, Items | Amazon DynamoDB, Ab-lambda-function-Lambda, ChatGPT

Address bar: ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#item-...

Navigation: Services, Search, [Alt+S]

Left sidebar: DynamoDB, Dashboard, Tables, Explore Items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations New, Reserved capacity, Settings, DAX, Clusters, Subnet groups

Tables (3): Any tag key, Any tag value, Find tables by table name, New-Product, Product, yash-r-table

Product: Autopreview, View table details

Scan or query items: Expand to query or scan items.

Items returned (1): Actions, Create item

	productid (String)	Category	product-name
	ASD	sample cate...	Sample product

Footer: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates. Privacy, Terms, Cookie preferences, 12:40, 06-03-2024